Spoken Digit Recognition with Wavelet Scattering and Deep Learning

The subject of this assignment is classification of spoken digits using following machine learning techniques.

After extracting features form spoken digits audio files using wavelet time scattering, these features are then used for training and testing with following networks:

- Support Vector Machine (SVM)
- Long Short-Term Memory (LSTM)
- LSTM optimized by Bayesian Optimization
- Deep Convolutional Neural Network (DCNN) with Mel-frequency Spectrograms

1. Dataset

Two datasets are used in this assignment:

- **Free Spoken Digit Dataset (FSDD)**

FSDD is an open data set consisting of 2000 recordings in English of the digits 0 through 9 obtained from four speakers. In this version, two of the speakers are native speakers of American English, one speaker is a nonnative speaker of English with a Belgian French accent, and one speaker is a nonnative speaker of English with a German accent. The data is sampled at 8000 Hz.

- **Student Spoken Digits Dataset**

This is a small dataset consisting of 50 recordings of digits 0 to 9 (10 recording per digit) all spoken by a non-native English speaker with an Iranian accent (myself).

2. Feature Extraction Using Wavelet Time Scattering

What a wavelet is?

Real world signals are mostly consisted of slowly changing trends or oscillations punctuated with some transients. We would focus on these changes as they provide interesting information but how can we extract this information? The first option would be FourierTransform but it cannot efficiently represent the time or position of abrupt changes. The fourier transform translates data as a sum of sine waves which oscillate forever and so are not localized in time or space.

Only functions that are well localized in time and frequency can accurately analyze signals with abrupt changes and wavelets have such charactersitics. A wavelet is a rapidly decaying wave like oscillation that has zero mean. Wavelets exist for a finite duration, and come in different sizes and shapes figure. Wavelets could be transformed in many ways inclusding scaling and shifting.

Wavelet Transforms can also be categorized as continious and discrete. Continious wavelet transforms are used for time frequency analysis and filtering of time localized frequency components; while discrete wavelet transforms are used for denoising and compression of signals and images.

Continuous wavelets scattering transform can be used as a powerful feature extractor for classification purposes. These extracted feature vectors will be used as the input to classifying neural networks.

3. Classifiers

3.1 Support Vector Machines (SVM) Classifier

Support Vector Machines algorithm is one of the most widely used supervised machine learning algorithms.

SVM results in higher accuracy with respect to other classifiers such as logistic regression, and decision trees. It is featured by the so called kernel trick for handling nonlinear input spaces. It cab be successfully used in a variety of problems such as face detection, and speech and handwriting recognition.

The SVM classifier aims to separate data points using an optimal hyperplane with the maximum margin. In order to define “optimal” hyperpalne, we need to introduce following elements of SVM (figure 1):

Hyperplane: a decision plane separating different classes.

Support Vectors: the nearest data points the hyperplane.

Margin: the perpendicular distance from the hyperplanes to the support vectors

SVM first generates several hyperplane in multidimensional space to separate different classes in an iterative manner. Among all generated hyperplanes, SVM selects the one with maximum margine and best classification accuracy.

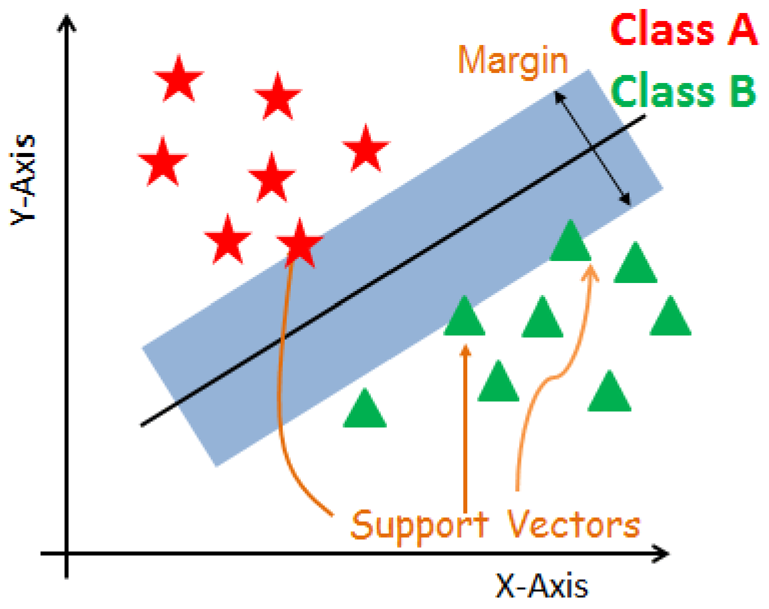


Figure 1: SVM Classification

3.2 Long Short-Term Memory (LSTM) Classifier

Recurrent Neural Networks and The Problem of Long-Term Dependencies

While feed-forward neural networks pass the data forward from input to output, recurrent neural networks have a feedback loop where data can be fed back into the input, allowing information to persist (figure 2). But they do not offer much control over the extent of preserving data but LSTM does.

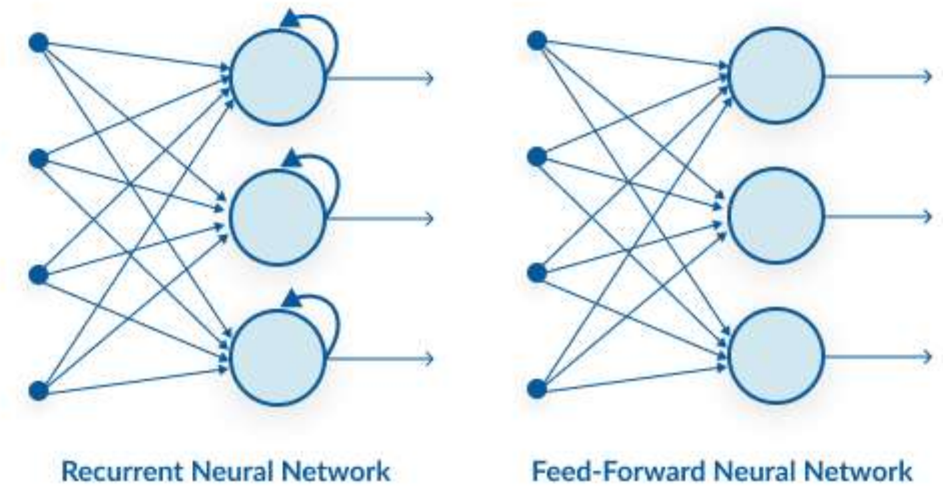


Figure 2: Recurrent NN vs. feed-forward NN

LSTM Networks

Long Short Term Memory networks – “LSTMs” – are a special kind of RNN, with the ability of learning long-term dependencies. LSTMs are particularly designed to remembering information for long periods of time but how?

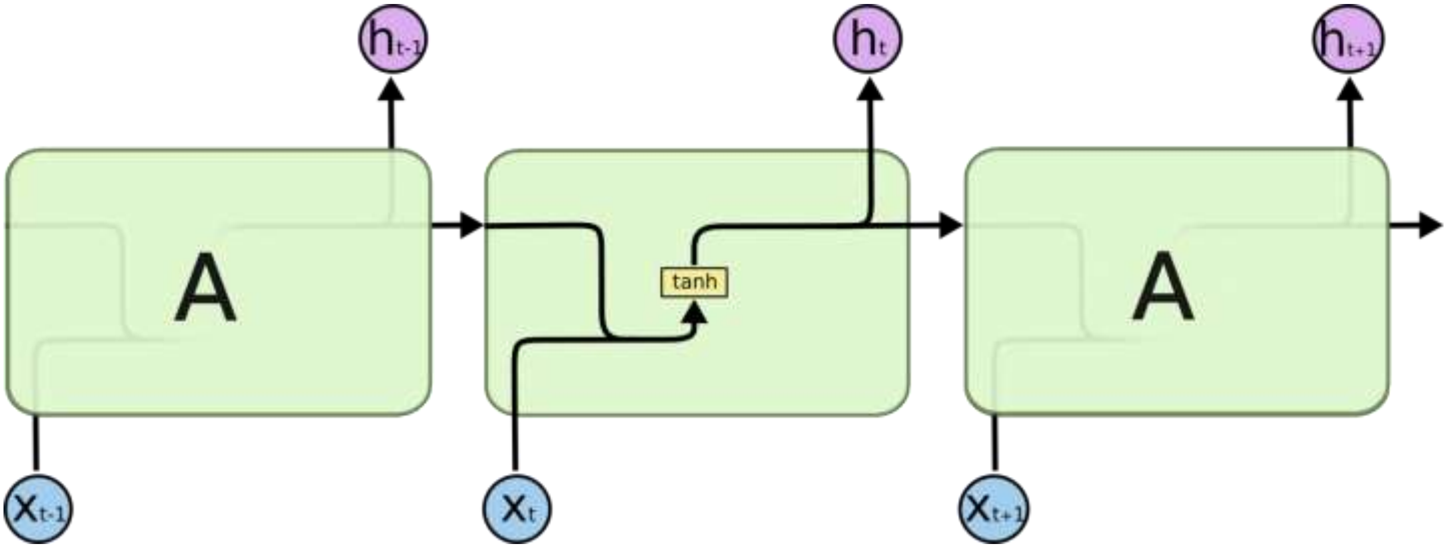


Figure 3: The repeating module in a standard RNN contains a single layer.

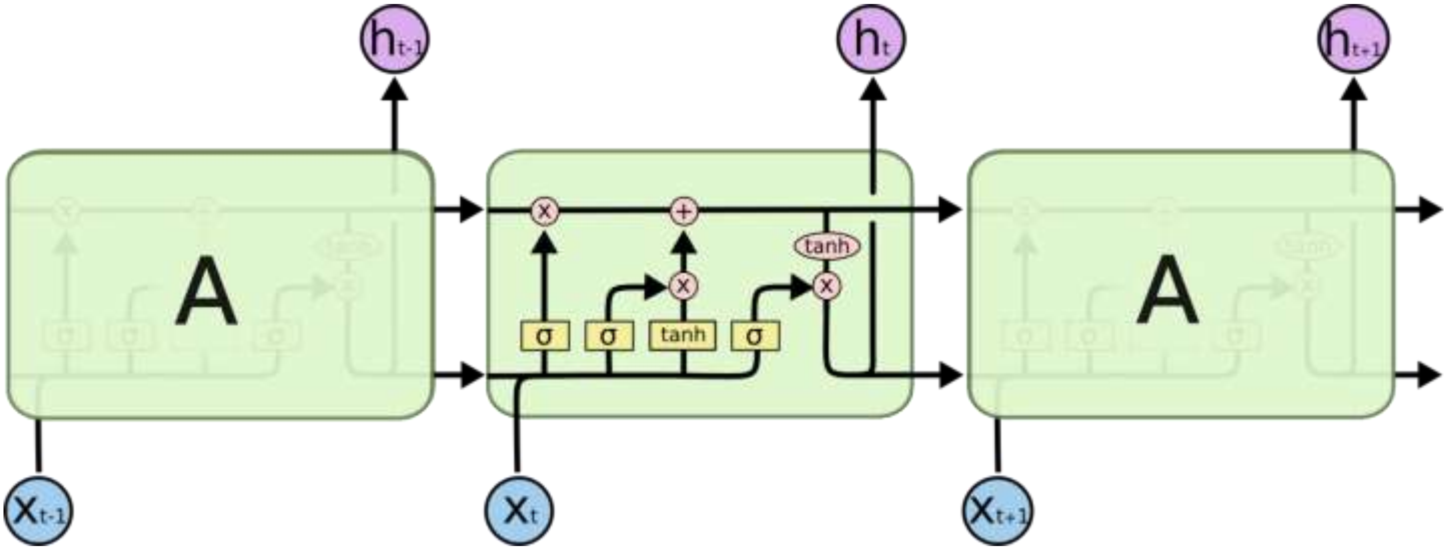


Figure 4: The repeating module in an LSTM contains four interacting layers.

Recurrent neural networks generally have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer (figure 3).

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way (figure 4).

The Core Idea Behind LSTMs

One main feature of LSTMs is the cell state, the horizontal line running through the top of the diagram. The cell state is similar to a conveyor belt, transferring the information.

Gates enable the LSTM to remove or add information to the cell state. Being composed of composed of a sigmoid neural net layer and a pointwise multiplication operation, gates optionally allow information through. An LSTM has three of these gates, to protect and control the cell state.

The sigmoid layer transforms inputs to outputs in the range of zero and one, regulating how much of each component should be let through. A value of zero means “let nothing through,” while a value of one means “let everything through!”

LSTM Architecture & Parameters

For this classification, we difine LSTM with following layers and parametes:

Table 1: LSTM Architecture & Parameters

No	Layer Type	Output Size	Details
1.	INPUT	321	-
2.	LSTM	512	Input Weights: 2048x321 Recurrent Weights: 2048x512 Bias: 2048x1
3.	FC	10	Weights: 10x512 Bias: 10x1
4.	SOFTMAX	10	-
5.	Class Output	1	-

3.3 LSTM optimized by Bayesian Optimization

Bayesian optimization is an optimization strategy for global optimization of expensive-to-evaluate. Since the objective function is not known, the Bayesian assumes a random function, placing a prior over it. The prior calculates beliefs about the behavior of the function. After quantifying the function evaluations, stored as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as infill sampling criteria) that determines the next query point.

In order to optimize LSTM, we need to tune two hyper parameters number of hidden units and initial learning rate in following ranges :

- Initial Learning Rate: [1e-5, 1e-1] => Optimal value = 2.199e-04
- Number of Hidden Units: [10, 1000] => Optimal value = 768

3.4 Deep Convolutional Neural Network (DCNN) Classifier

The last classifier is a DCNN that we define a DCNN with following architecture and parameters:

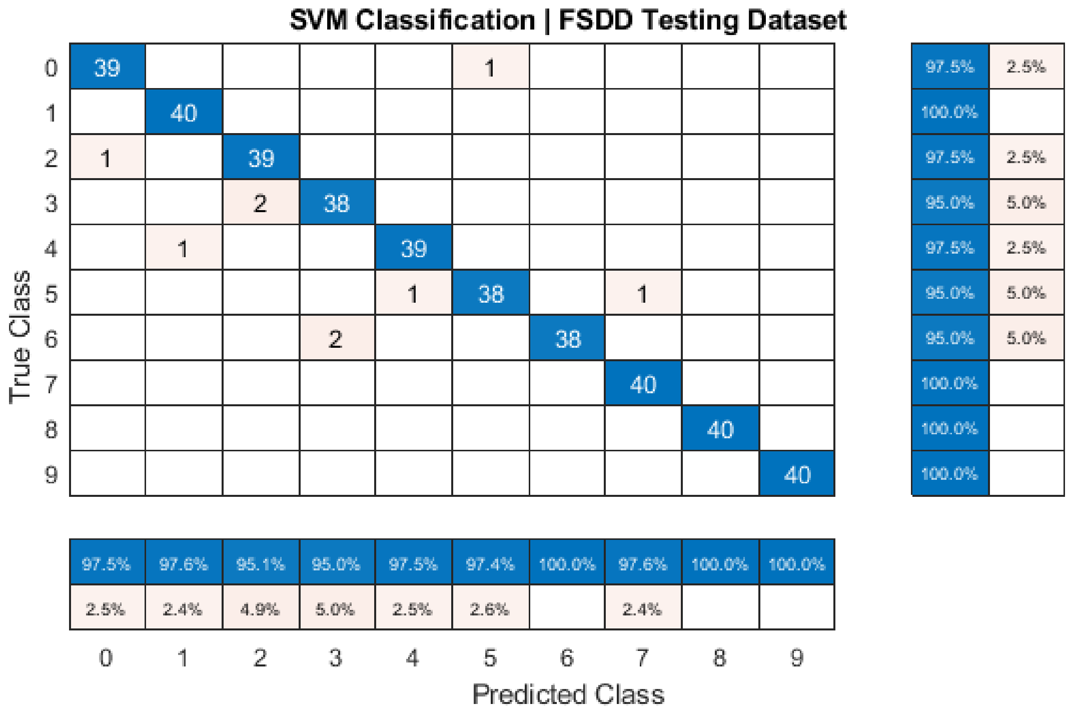
Table 2: DCNN Architecture & Parameters

No	Layer Type	Output Size	Filter Size / Stride
1.	INPUT IMAGE	40×81×1	
2.	CONV	40×81×12	5×5;K = 12
3.	BN	40×81×12	
4.	ACT (ReLU)	40×81×12	
5.	POOL	20×41×12	F=3, S=2
6.	CONV	20×41×24	3x3;K = 24
7.	BN	20×41×24	
8.	ACT (ReLU)	20×41×24	
9.	POOL	10×21×24	F=3, S=2
10.	CONV	10×21×48	3x3;K = 48
11.	BN	10×21×48	
12.	ACT (ReLU)	10×21×48	
13.	POOL	5×11×48	F=3, S=2
14.	CONV	5×11×48	3x3;K = 48
15.	BN	5×11×48	
16.	ACT (ReLU)	5×11×48	
17.	CONV	5×11×48	3x3;K = 48
18.	BN	5×11×48	
19.	ACT (ReLU)	5×11×48	
20.	POOL	4×10×48	F=2, S=1
21.	DO	4×10×48	dropoutProb = 0.2
22.	FC	1x1x10	
23.	SOFTMAX	1x1x10	
24.	Class Output	1x1x1	

4. Results & Discussion

4.1 Support Vector Machine (SVM) Classifier

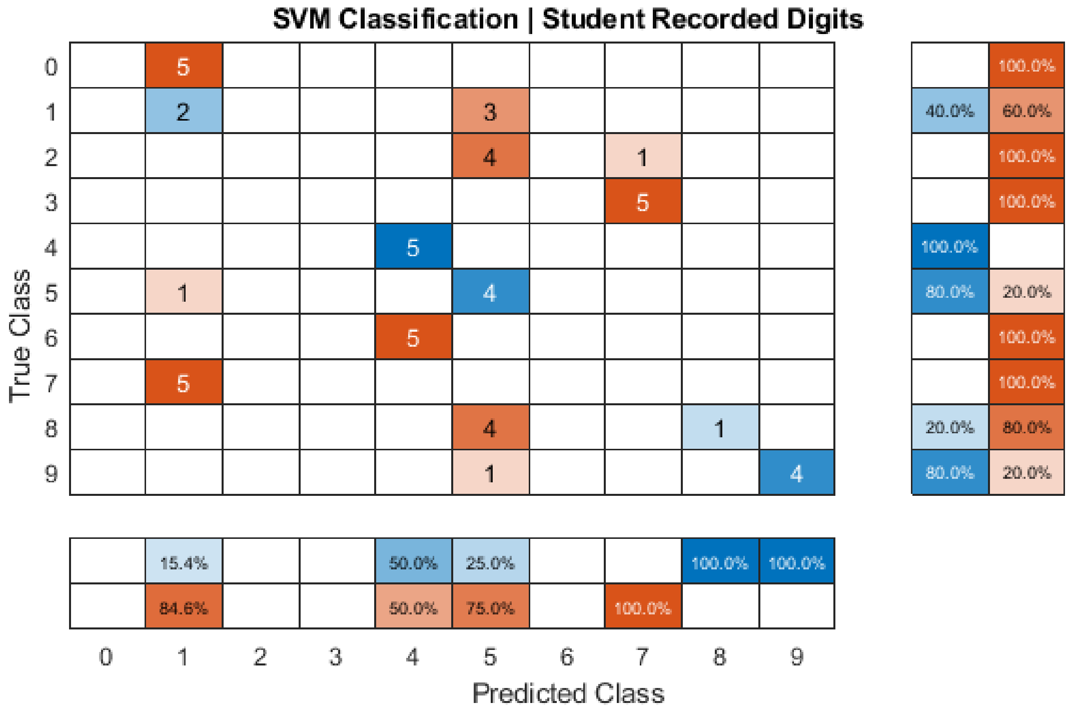
4.1.1 SVM Training & Testing on FSDD Dataset



True Class	0	1	2	3	4	5	6	7	8	9		
0	39					1					97.5%	2.5%
1		40									100.0%	
2	1		39								97.5%	2.5%
3			2	38							95.0%	5.0%
4		1			39						97.5%	2.5%
5					1	38		1			95.0%	5.0%
6				2			38				95.0%	5.0%
7								40			100.0%	
8									40		100.0%	
9										40	100.0%	
	97.5%	97.6%	95.1%	95.0%	97.5%	97.4%	100.0%	97.6%	100.0%	100.0%		
	2.5%	2.4%	4.9%	5.0%	2.5%	2.6%		2.4%				

Figure 5: SVM Training & Testing on FSDD Dataset

4.1.2 SVM Evaluation on Student Spoken Digits



True Class	0	1	2	3	4	5	6	7	8	9		
0		5										100.0%
1		2				3					40.0%	60.0%
2						4		1				100.0%
3								5				100.0%
4					5						100.0%	
5		1				4					80.0%	20.0%
6					5							100.0%
7		5										100.0%
8						4			1		20.0%	80.0%
9						1				4	80.0%	20.0%
		15.4%			50.0%	25.0%			100.0%	100.0%		
		84.6%			50.0%	75.0%		100.0%				

Figure 6: SVM Evaluation on Student Spoken Digits

4.1.3 SVM Summary

Classifier: SVM

Specification: quadratic polynomial kernel

Feature vector: N x 321 x 1

Training dadaset: FSDD (N=160 x 10)

Testing dadaset: FSDD (N=40 x 10) => test accuracy= 97.75%

Evaluation dadaset: Student (N=5 x 10) => test accuracy= 32%

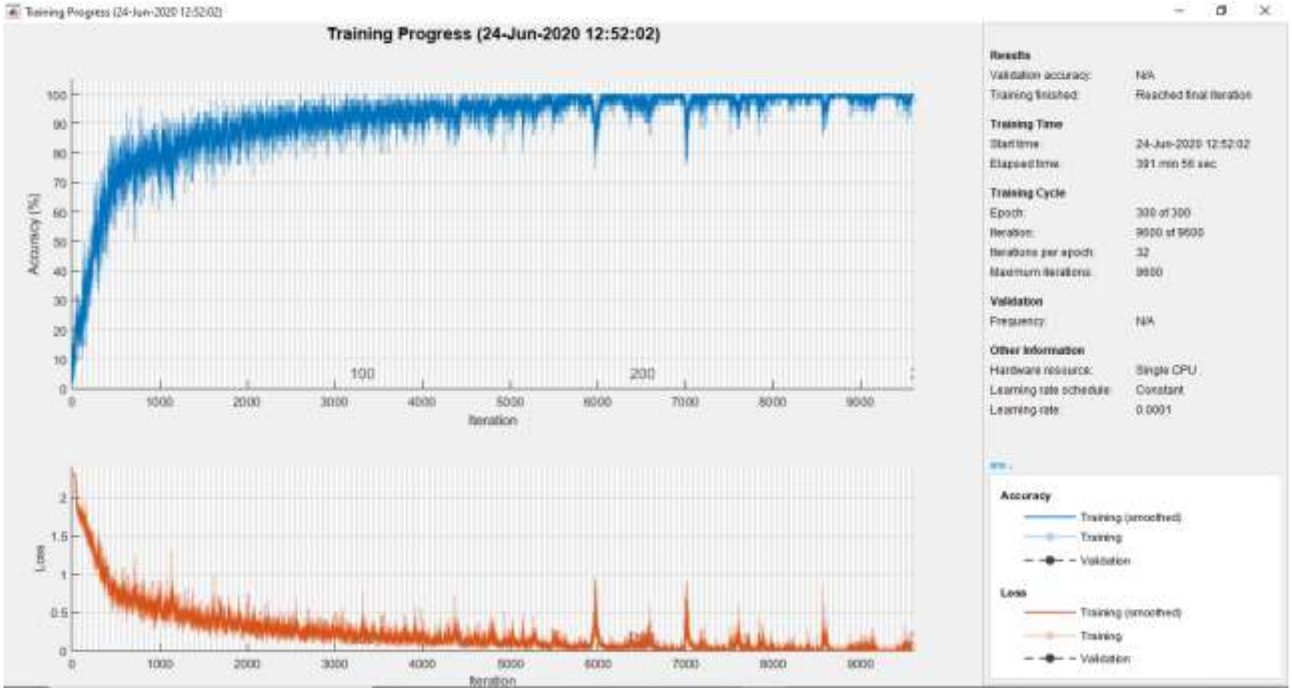
Remarks:

SVM classifies DSDD dataset with an excellent accuracy of 98%; however, its classification accuracy for student spoken digits is significantly lower, 32%.

It is worth mentioning that the classification accuracy of different digits are very distict, ranging from 0% to 100%.

4.2 Long Short-Term Memory (LSTM) Classifier

4.2.1 LSTM Training on FSDD Dataset



Results

Validation accuracy: N/A

Training finished: Reached final iteration

Training Time

Start time: 24-Jun-2020 12:52:02

Elapsed time: 391 min 56 sec

Training Cycle

Epoch: 300 of 300

Iteration: 9600 of 9600

Iterations per epoch: 32

Maximum iterations: 9600

Validation

Frequency: N/A

Other Information

Hardware resource: Single CPU

Learning rate schedule: Constant

Learning rate: 0.0001

Accuracy

Training (smoothed)

Training

Validation

Loss

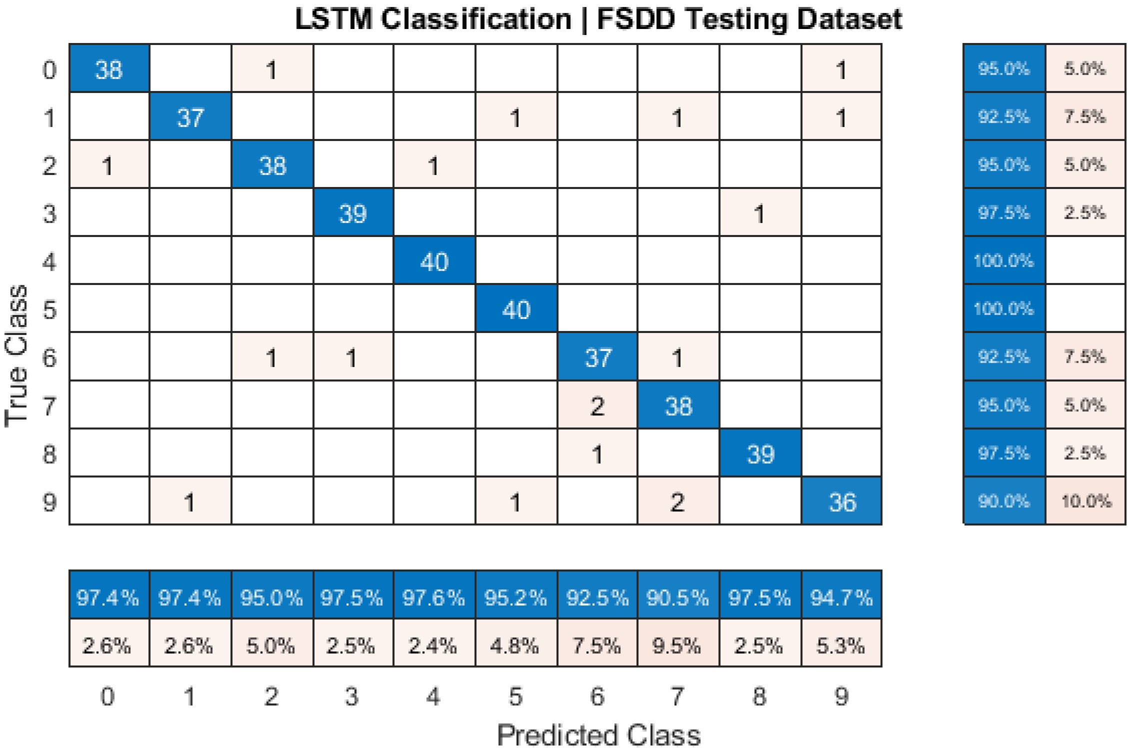
Training (smoothed)

Training

Validation

Figure 7: LSTM Training on FSDD Dataset

4.2.2 LSTM Testing on FSDD Dataset



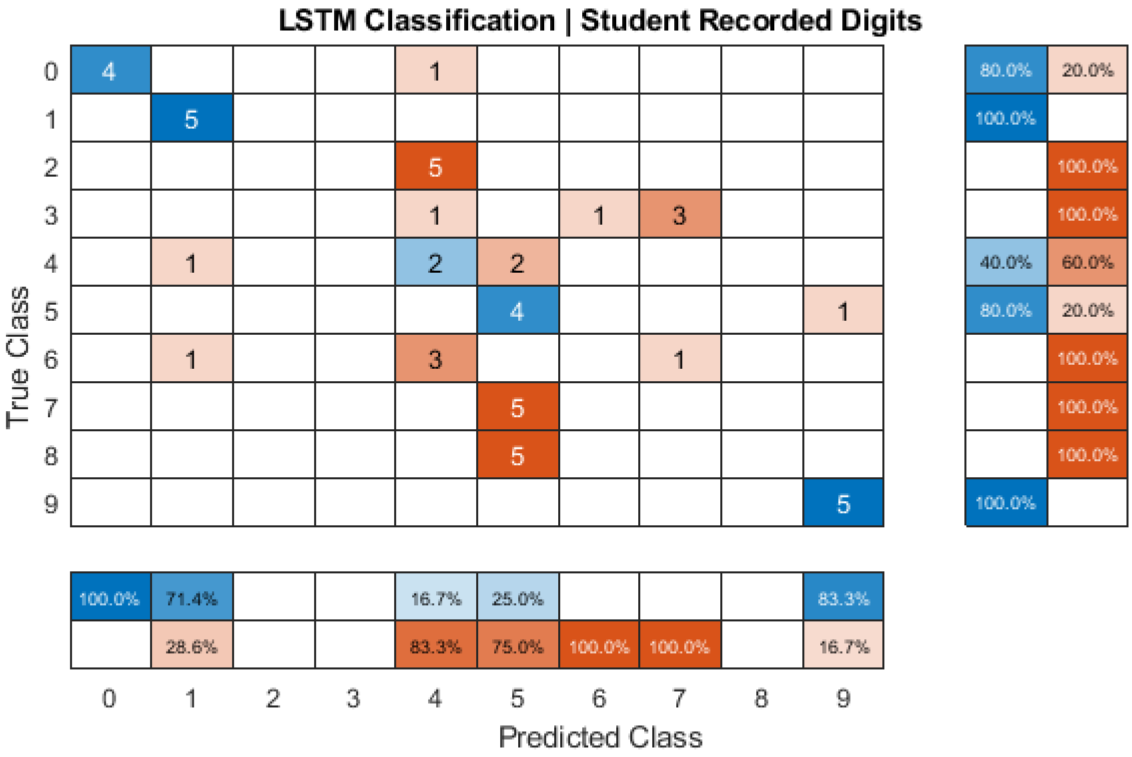
LSTM Classification | FSDD Testing Dataset

True Class	0	1	2	3	4	5	6	7	8	9		
0	38		1							1	95.0%	5.0%
1		37				1		1		1	92.5%	7.5%
2	1		38		1						95.0%	5.0%
3				39					1		97.5%	2.5%
4					40						100.0%	
5						40					100.0%	
6			1	1			37	1			92.5%	7.5%
7							2	38			95.0%	5.0%
8							1		39		97.5%	2.5%
9		1				1		2		36	90.0%	10.0%
	97.4%	97.4%	95.0%	97.5%	97.6%	95.2%	92.5%	90.5%	97.5%	94.7%		
	2.6%	2.6%	5.0%	2.5%	2.4%	4.8%	7.5%	9.5%	2.5%	5.3%		
	0	1	2	3	4	5	6	7	8	9		

Predicted Class

Figure 8: LSTM Testing on FSDD

4.2.3 LSTM Evaluation on Student Spoken Digits



	0	1	2	3	4	5	6	7	8	9		
0	4				1						80.0%	20.0%
1		5									100.0%	
2					5							100.0%
3					1		1	3				100.0%
4		1			2	2					40.0%	60.0%
5						4				1	80.0%	20.0%
6		1			3			1				100.0%
7						5						100.0%
8						5						100.0%
9										5	100.0%	

0	1	2	3	4	5	6	7	8	9
100.0%	71.4%			16.7%	25.0%				83.3%
	28.6%			83.3%	75.0%	100.0%	100.0%		16.7%

Figure 9: LSTM Evaluation on Student Spoken Digits

4.2.4 LSTM Summary:

Classifier: LSTM

Specification: 5-layer: INPUT=>LSTM=>FC=>SOFTMAX=>OUTPUT

512 hidden units, Initial Learning Rate: 0.0001

Feature vector: N x 321 x 1

Training dadaset: FSDD (N=160 x 10)

Testing dadaset: FSDD (N=40 x 10) => test accuracy= 94%

Evaluation dadaset: Student (N=5 x 10) => test accuracy= 40%

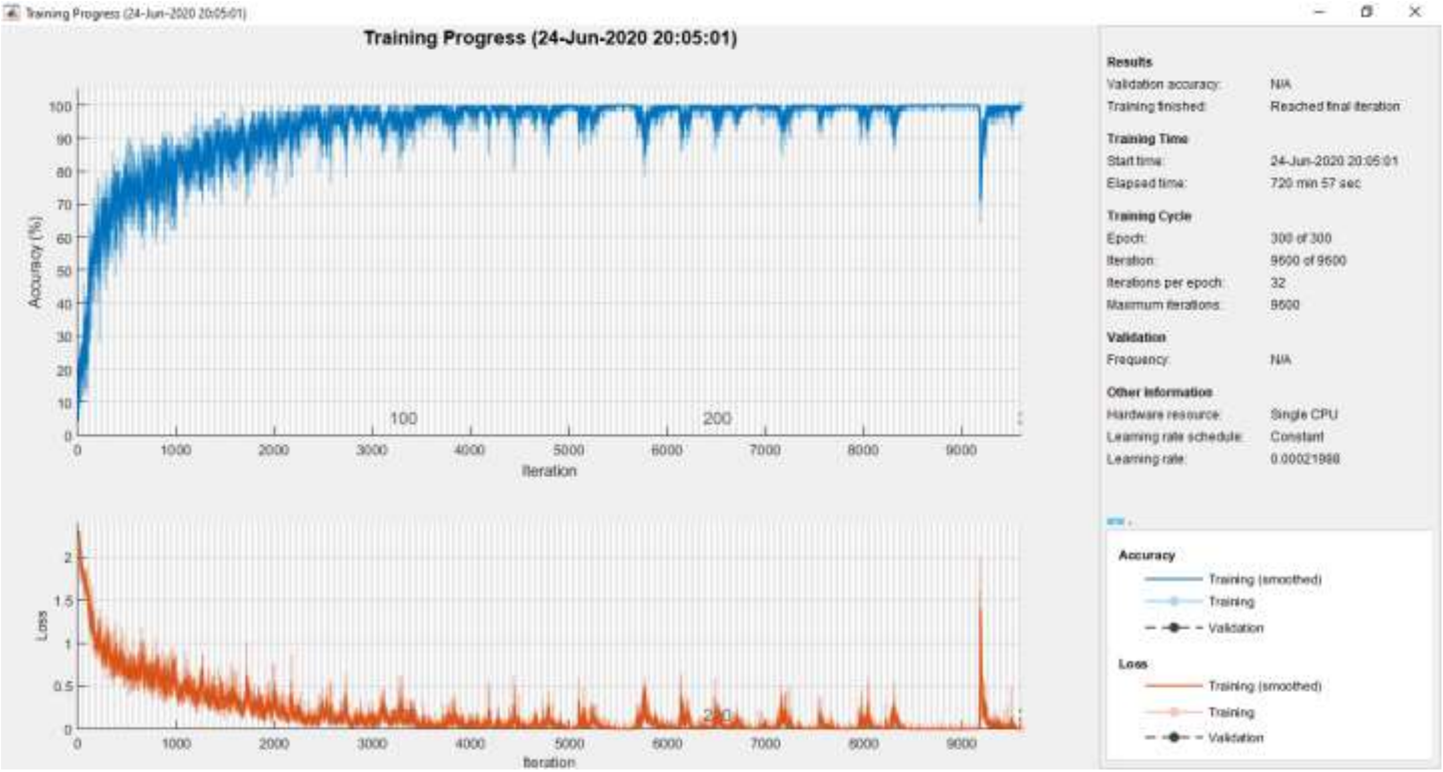
Remark:

LSTM classifies FSDD dataset with an excellent accuracy of 94%; however, its classification accuracy for student spoken digits is significantly lower, 40%.

It is worth mentioning that the classification accuracy of different digits are very distict, ranging from 0% to 100%.

4.3 LSTM-Optimized by Bayesian Optimization

4.3.1 LSTM-Optimized Training on FSDD Dataset



Training Progress (24-Jun-2020 20:05:01)

Results

Validation accuracy:	N/A
Training finished:	Reached final iteration

Training Time

Start time:	24-Jun-2020 20:05:01
Elapsed time:	720 min 57 sec

Training Cycle

Epoch:	300 of 300
Iteration:	9600 of 9600
Iterations per epoch:	32
Maximum iterations:	9600

Validation

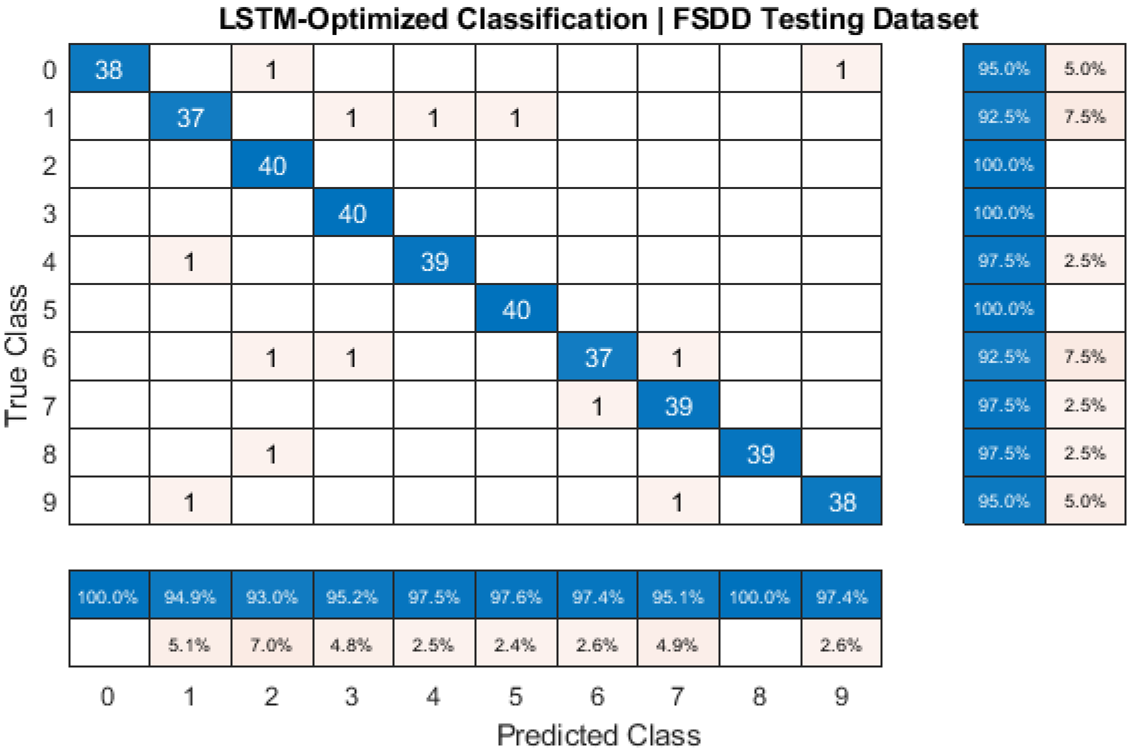
Frequency:	N/A

Other information

Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	0.00021988

Figure 10: LSTM-Optimized Training on FSDD Dataset

4.3.2 LSTM-Optimized Testing on FSDD Dataset



LSTM-Optimized Classification | FSDD Testing Dataset

True Class	0	1	2	3	4	5	6	7	8	9		
0	38		1							1	95.0%	5.0%
1		37		1	1	1					92.5%	7.5%
2			40								100.0%	
3				40							100.0%	
4		1			39						97.5%	2.5%
5						40					100.0%	
6			1	1			37	1			92.5%	7.5%
7							1	39			97.5%	2.5%
8			1						39		97.5%	2.5%
9		1						1		38	95.0%	5.0%
	100.0%	94.9%	93.0%	95.2%	97.5%	97.6%	97.4%	95.1%	100.0%	97.4%		
		5.1%	7.0%	4.8%	2.5%	2.4%	2.6%	4.9%		2.6%		

Predicted Class

Figure 11: LSTM-Optimized Testing on FSDD Dataset

4.3.3 LSTM-Optimized. Evaluation on Student Spoken Digits

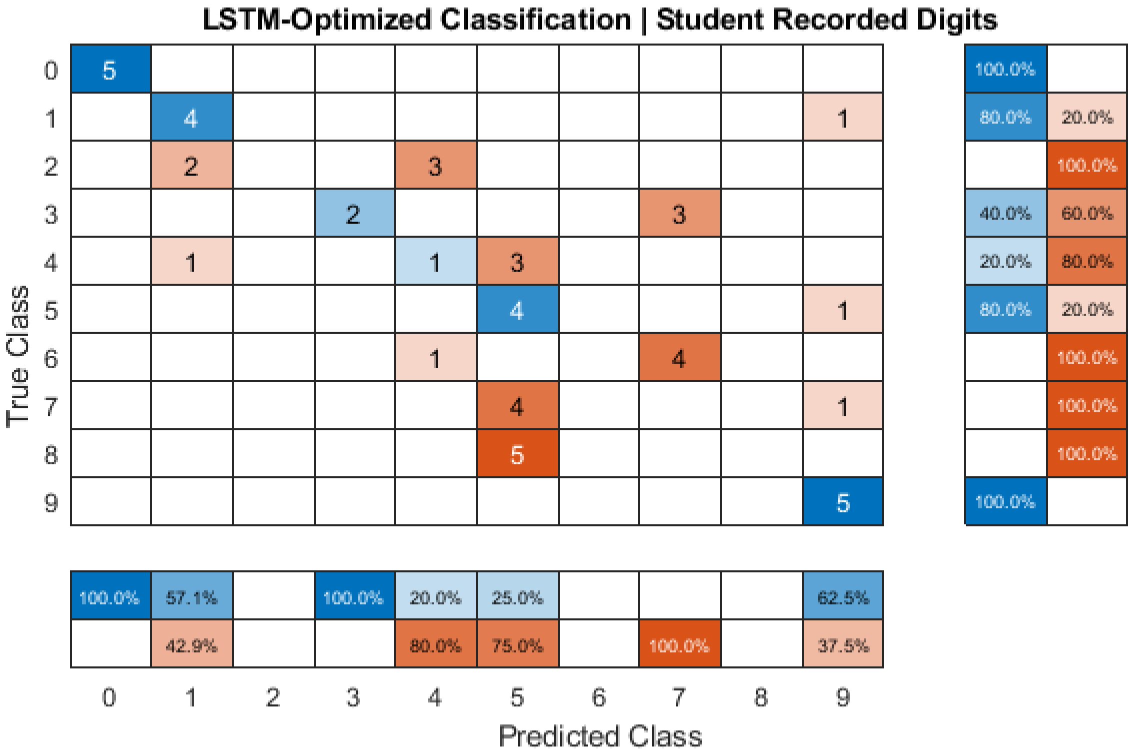


Figure 12: LSTM-Optimized. Evaluation on Student Spoken Digits

4.3.4 LSTM- Optimized Summary:

Classifier: LSTM-Optimized

Optimization Parameters and range

Initial Learning Rate: [1e-5, 1e-1] => **Optimal value** = 2.1988e-4

Number of Hidden Units: [10, 1000] => **Optimal value** = 768

Feature vector: N x 321 x 1

Training dadaset: FSDD (N=160 x 10)

Testing accuracy: FSDD (N=40 x 10) => test accuracy: 94% increased to 97.75%

Evaluation dadaset: Student (N=5 x 10) => test accuracy: 40% increased to 42%

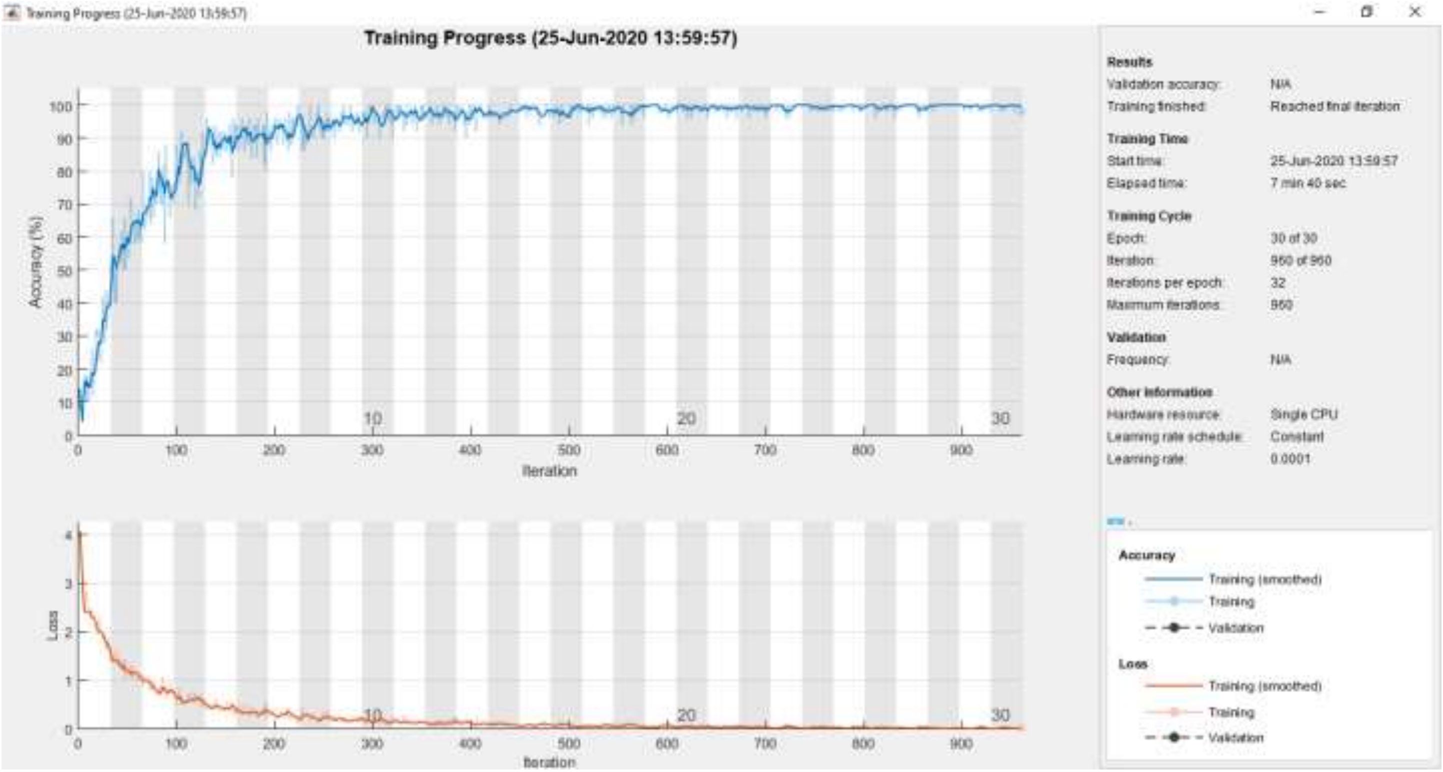
Remarks:

Bayes optimization increased testing accuracy from 94% to 97.75%, and evaluation accuracy also increased by 2%.

The increase percentage is not significant which shows the initial values were already optimized.

4.4 Deep Convolutional Neural Network (DCNN)

4.4.1 DCNN Training on FSDD Dataset



Results
Validation accuracy: N/A
Training finished: Reached final iteration

Training Time
Start time: 25-Jun-2020 13:59:57
Elapsed time: 7 min 40 sec

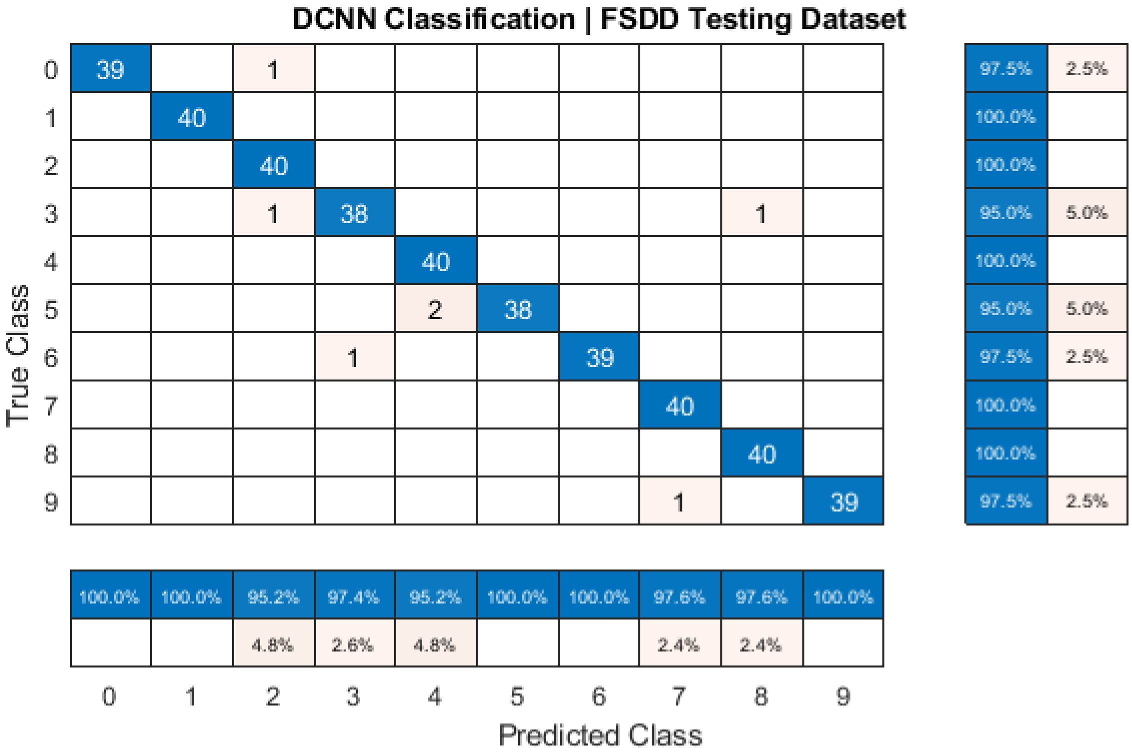
Training Cycle
Epoch: 30 of 30
Iteration: 960 of 960
Iterations per epoch: 32
Maximum iterations: 960

Validation
Frequency: N/A

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Figure 13: DCNN Training on FSDD Dataset

4.4.2 DCNN Testing on FSDD Dataset

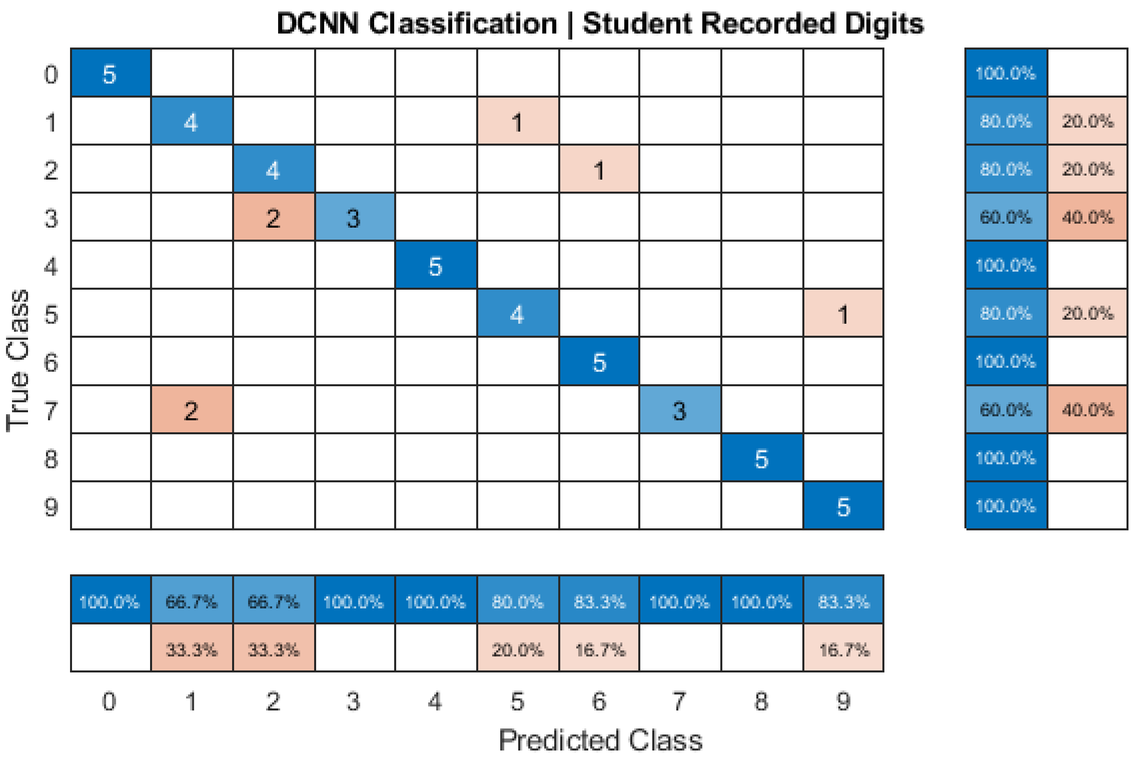


DCNN Classification | FSDD Testing Dataset

	0	1	2	3	4	5	6	7	8	9		
0	39		1								97.5%	2.5%
1		40									100.0%	
2			40								100.0%	
3			1	38					1		95.0%	5.0%
4					40						100.0%	
5					2	38					95.0%	5.0%
6				1			39				97.5%	2.5%
7								40			100.0%	
8									40		100.0%	
9								1		39	97.5%	2.5%
	100.0%	100.0%	95.2%	97.4%	95.2%	100.0%	100.0%	97.6%	97.6%	100.0%		
			4.8%	2.6%	4.8%			2.4%	2.4%			

Figure 14: DCNN Testing on FSDD Dataset

4.4.3 DCNN Evaluation on Student Spoken Digits



	0	1	2	3	4	5	6	7	8	9		
0	5										100.0%	
1		4				1					80.0%	20.0%
2			4				1				80.0%	20.0%
3			2	3							60.0%	40.0%
4					5						100.0%	
5						4				1	80.0%	20.0%
6							5				100.0%	
7		2						3			60.0%	40.0%
8									5		100.0%	
9										5	100.0%	
	100.0%	66.7%	66.7%	100.0%	100.0%	80.0%	83.3%	100.0%	100.0%	83.3%		
		33.3%	33.3%			20.0%	16.7%			16.7%		

Figure 15: DCNN Evaluation on Student Spoken Digits

4.4.4 DCNN Summary:

Classifier: DCNN

Specification: 24-layer

Feature vector: 40 x 81 x 1 x N

Training dadaset: FSDD (N=160 x 10)

Testing dadaset: FSDD (N=40 x 10) => test accuracy= 98%

Evaluation dadaset: Student (N=5 x 10) => test accuracy= 86%

Remark:

DCNN classifies FSDD dataset with a slightly higher accuracy of 98%. Its strength is revealed when it can improve the classification accuracy for student spoken digits from 40% to 86%. This means that not only DCNN is stronger than other networks.

4.5 Classifier Summary

The overall performances of classifiers is summarized in folloing table. Following points can be concluded form this table:

- SVM and LSTIM perform excellently on FDSS dataset but poorly on student dataset
- Bayes optimization increased accuracy by 2-4% as hyper parameters are already close to optimal values
- DCNN outperforms other classifiers by performing excellent not only on FSDD dataset but also on student dataset

Table 3: Classifier Performance

Classifier	Classification Accuracy [FDSS Dataset]	Classification Accuracy [Student Dataset]
SVM	97.75%	32%
LSTM	94%	40%
LSTM-Optimized	97.75%	42%
DCNN	98.25%	86%

References

[1] https://www.mathworks.com/help/audio/examples/spoken-digit-recognition-with-wavelet-scattering-and-deep-learning.html

[2] Free Spoken Digit Dataset (FSDD), available at https://github.com/Jakobovski/free-spoken-digit-dataset

[3] https://www.mathworks.com/videos/series/understanding-wavelets-121287.html

[4] https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

[5] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[6] https://en.wikipedia.org/wiki/Bayesian_optimization#:~:text=Bayesian%20optimization%20is%20a%20sequential,expensive%2Dto%2Devaluate%20functions.